Chapter

3

BEYOND THE BOX

A practical approach to quality Project Process

Determining Sufficient Standards

Defining and Applying Sufficient Standards

(Determining Threshold of Acceptable Project Risk)

Bar Biszick, CSQA (11/1/02)

Issue: How do we determine “Sufficient Standards”?

Standards are about defining corporate risk tolerance levels for work productivity and quality. Most companies defer standards definition and enforcement to its line management staff and workers. The question is: do these standards or the absence of them meet management's tolerance level for productivity, cost and product quality?

Objective:

This proposal offers a low cost means of determining Project Standards Sufficiency for a given project team. By applying this method, management can identify product and workflow insufficiencies that can be traced directly to cost overruns, schedule delays and interpersonal team issues. This approach is part of the Beyond The Box Methodology, a set of quality practices designed to adapt proven quality practices rooted in the Quality Assurance Institute’s Common Body of Knowledge that supports industry accepted standards including Total Quality Management, SEI’s capability maturity model and ISO 9000 standards, to dynamic technology projects. It introduces Microsoft Solutions Framework and CORE principles that stress the need to consider the meta triangle of Industry Dynamics, Corporate Mission, and Personal Politics impacting the Project Triangle for any significant cultural change.

Strategy:

1) Compare the attributes and perceived insufficiencies of two comparable projects done by the same group and determine their impacts to budget and quality of product/service.
2) Let the evaluated intersections between the two projects define the baseline of performance for the group, and articulate it with metrics.
3) Identify the root cause of insufficiencies (ie. Communication problems, inappropriate skill set, insufficient hardware in the dev environment etc.)
4) Work with manager to begin raising awareness of baseline with their team.
5) Implement controls to raise insufficiencies to baseline levels.
6) Continue measuring and monitoring projects going forward.

Terminology:

The term "bug" historically means a defect in the product. For this discussion, we need to expand our notion of defects to encompass both product and process. An issue, therefore, is any activity, event or deliverable that impedes the team’s expected productivity. In addition to product defects, this might include the quality of documentation needed for subsequent phases, misunderstandings over scope and schedule, changing Requirements or miscommunications, dev or test environment management or materials insufficiencies, timely access to team resources, information or decisions, accidental oversights in task interpretation etc.

Who measures:

QA, because it is least burdened throughout most of the project, the most skilled, most objective, and most affected by workflow process issues and interim deliverables defects.

What gets measured:

Any issue or event that impedes expected productivity.

- Issues raised concerning any interim deliverable on which the project and testing depends
- Workflow issues (team communication, access to information, schedule, timeliness of events...)

Tools:

Beyond the Box Test Project Tracking Template
Beyond the Box Testable Statements Template

Defining and Applying Sufficient Standards—

Real World Example:

This technique was applied to two projects completed by the same personnel. Here is the progress and results.

Step 1: Compare the attributes and perceived insufficiencies of two comparable projects done by the same group and determine their impacts to budget and quality of product/service.

Project Equivalencies:

Same group, same PM
Testing took the same amount of time and took place over same calendar period (approx. 6 weeks)
All tests were evaluated each cycle on both projects. Those ready for testing were run.
Same number of personnel involved in both projects
All key roles involved from start to finish.
QA achieved >95% coverage/completion on both projects by the last cycle.
Found roughly equivalent number of bugs.
Approximately the same number of issues were CLOSED in both projects.
Exactly the same number of items were evaluated on both projects.

Potentially Significant differences:

Different evaluators. However, both evaluators reviewed and approved each other's work.
4 more bugs were found in Project A than in Project B
3 more issues were postponed in Project A than in Project B
Based on test case counts Project A is 2.49% more complex a system than Project B which may account for why
37% more hours were needed to test 22% more tests in Project A.
Both projects contributed to calendar date delay that was double the expected calendar time

Problem Definition:

Project A exceeded QA budget estimate by 11% whereas Project B exceeded QA estimate by 53%

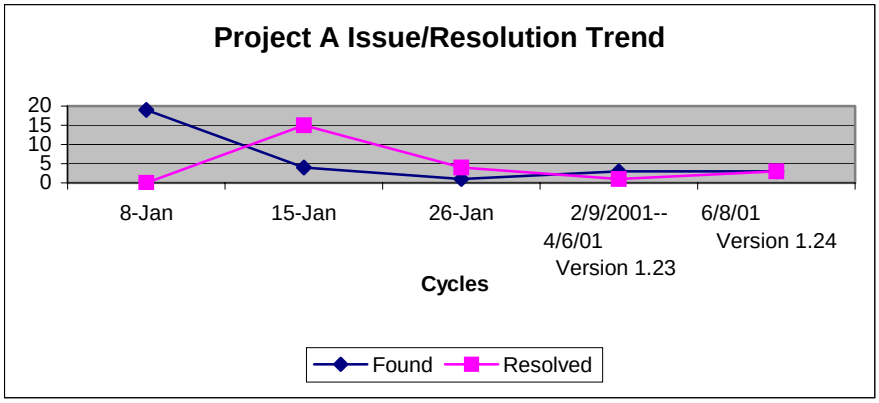
Project Descriptions/Relationship

A production critical bug was found that could potentially post duplicate or incomplete data to a back end system. Targeted testing for the Project A application and Project B application was undertaken to confirm correct functioning of an upgrade to a file that was common to both applications. Each project also added new functionality and refined or fixed existing production code.

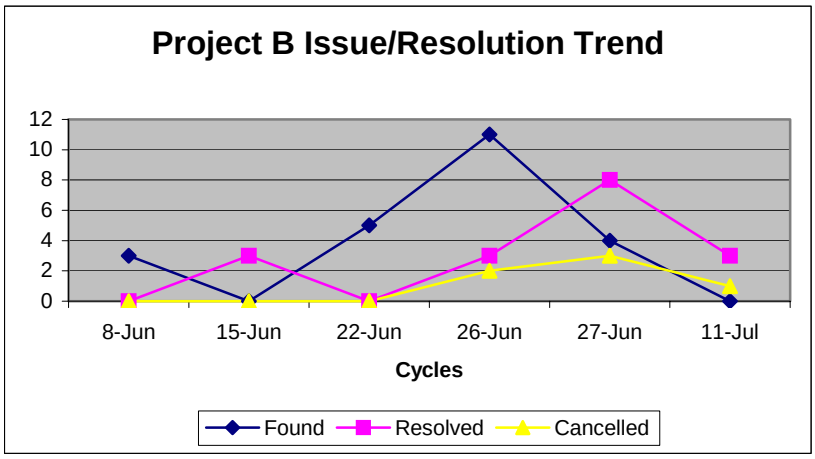
What can metrics tell us?

Issue Trends:

Project A represents and ideal issue trend, where most of the bugs are banged out in the first cycle of testing. Then few issues are found and resolved in following cycles.



Project B shows highest % of issues late in cycle testing. This might indicate dev was still coding instead of fixing, Requirements were still in flux, QA misunderstood what was required for testing.



Issue Breakout

- System Design/Coding Issues accounted for 95% of Project A issues as compared to 0% on Project B. This might indicate differences in coder skill/experience, scope of design/coding work or in system complexity etc.
- Process and Test Dependency Issues accounted for 0% of Project A issues as compared to 75% of Project B issues, where 43% of all issues were found during Build Migration. This might indicate a problem with source control, installation package or documentation, Change communication, or Test Lab Environment management.

	Project A		Project B	
Build Verification & Smoke Testing	0	0%	10	43%
Traceability	0	0%	6	26%
Acceptance and Usability	4	15%	3	13%
Internal Functions and Transactions	5	19%	0	0%
Configuration/Integration/Interdependencies	4	15%	0	0%
GUI-View Panel/User Interactions	15	56%	0	0%
Security	0	0%	0	0%
Test Dependencies	0	0%	2	9%
User Inputs	0	0%	0	0%
Business Continuity	2	7%	2	9%

Test Completion trend

Testing was 95% complete in the Project A penultimate cycle but only 82% done in Project B. This supports the idea that during the Project B penultimate cycle, there were still test blockage or unknowns that impeded full cycle testing.

Project A	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5
Completion	83%	79%	84%	95%	97%

Project B	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6
Completion	66%	73%	76%	80%	81%	95%

Test Areas/Impacts

Of those areas that fell below 60% completion, only Traceability on Project B had an issue impact.
Traceability is verification of the Business Requirements through all subsequent deliverables on into the product. This might indicate a problem with the Requirements definition, refinement, fluxuation, team shared understanding etc…

Project B	Defects	%	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6
Traceability	6	26%	33%	33%	33%	33%	44%	44%
Security	0	0%	0%	0%	0%	0%	0%	25%

Testing Cost Overruns

Project A ran 11% over budget (extra 7 hours) whereas Project B cost more than double the estimate. (extra 51 hours)

Project A		
IBudgeted	Actual	% Deviation
60	67	10%

Project B		
Budgeted	Actual	% Deviation
45	96	53%

What Cost QA the Most?

Project B Life Cycle	Budgeted	Actual	% Done
Execute and Report System Tests	**16.5**	**45**	**272.73%**
Requirements Analysis			
Build Verification Tests/Smoke Tests	1.5	10	666.67%
Project Change Control/Communications Plan	0.1	4	4000.00%
Dev Change Control/Communications Review	0.1	4	4000.00%
Test Requirements	1	4	400.00%

What had the highest estimate overrun?

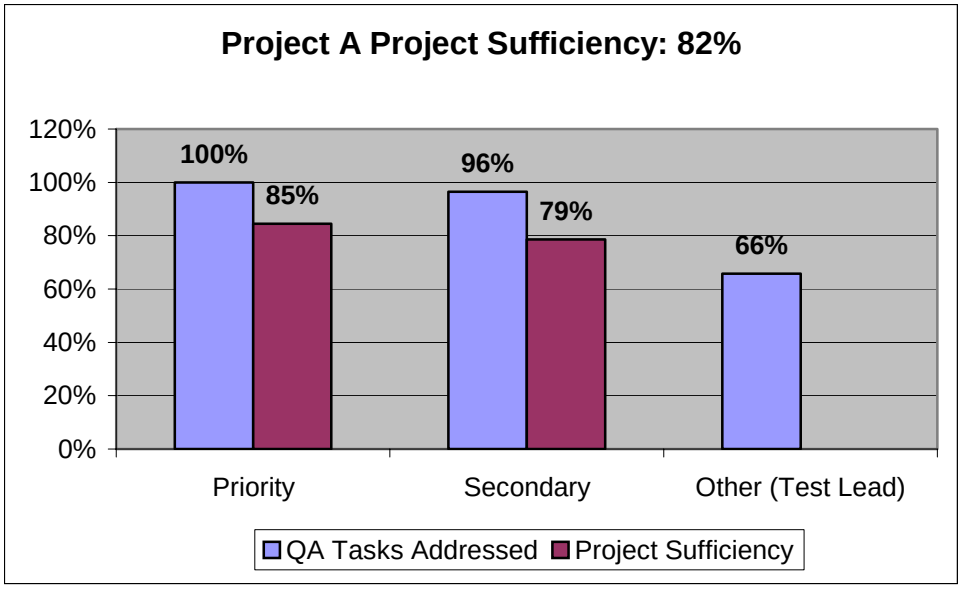
These items accounted for an extra 20 hours in QA time, where Build Verification and Smoke Testing accounts for 50% of these extra costs
and Change Communication accounts for 40%.

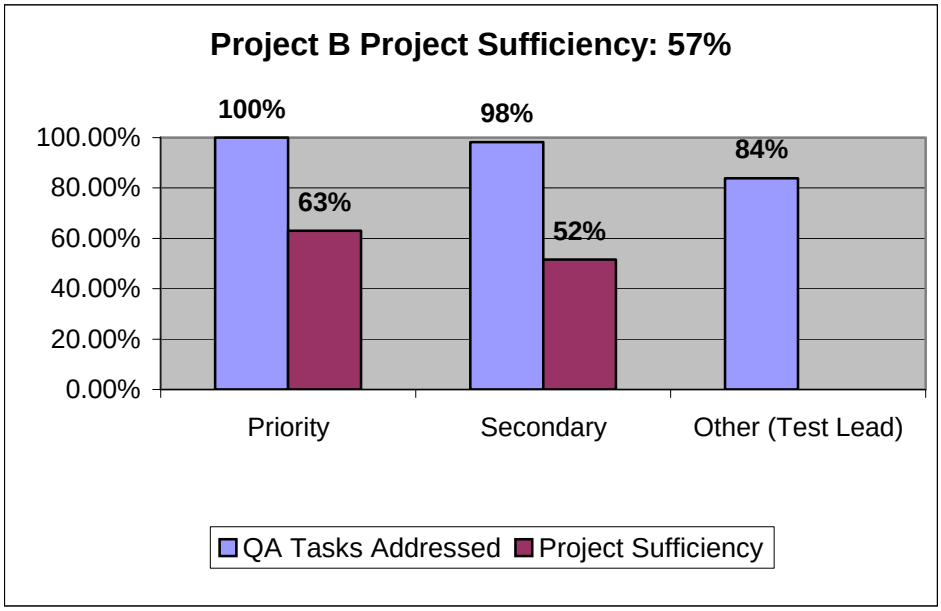
Project B Life Cycle	Budgeted	Actual	% Done
Project Change Control/Communications Plan	0.1	4	4000.00%
Dev Change Control/Communications Review	0.1	4	4000.00%
Compliance: Business Rules	0.1	1	1000.00%
Risk Mitigation Review	0.1	1	1000.00%
Build Verification Tests/Smoke Tests	1.5	10	666.67%

Step 2: Let the evaluated intersections between the two projects define the baseline of performance for the group, and articulate it with metrics

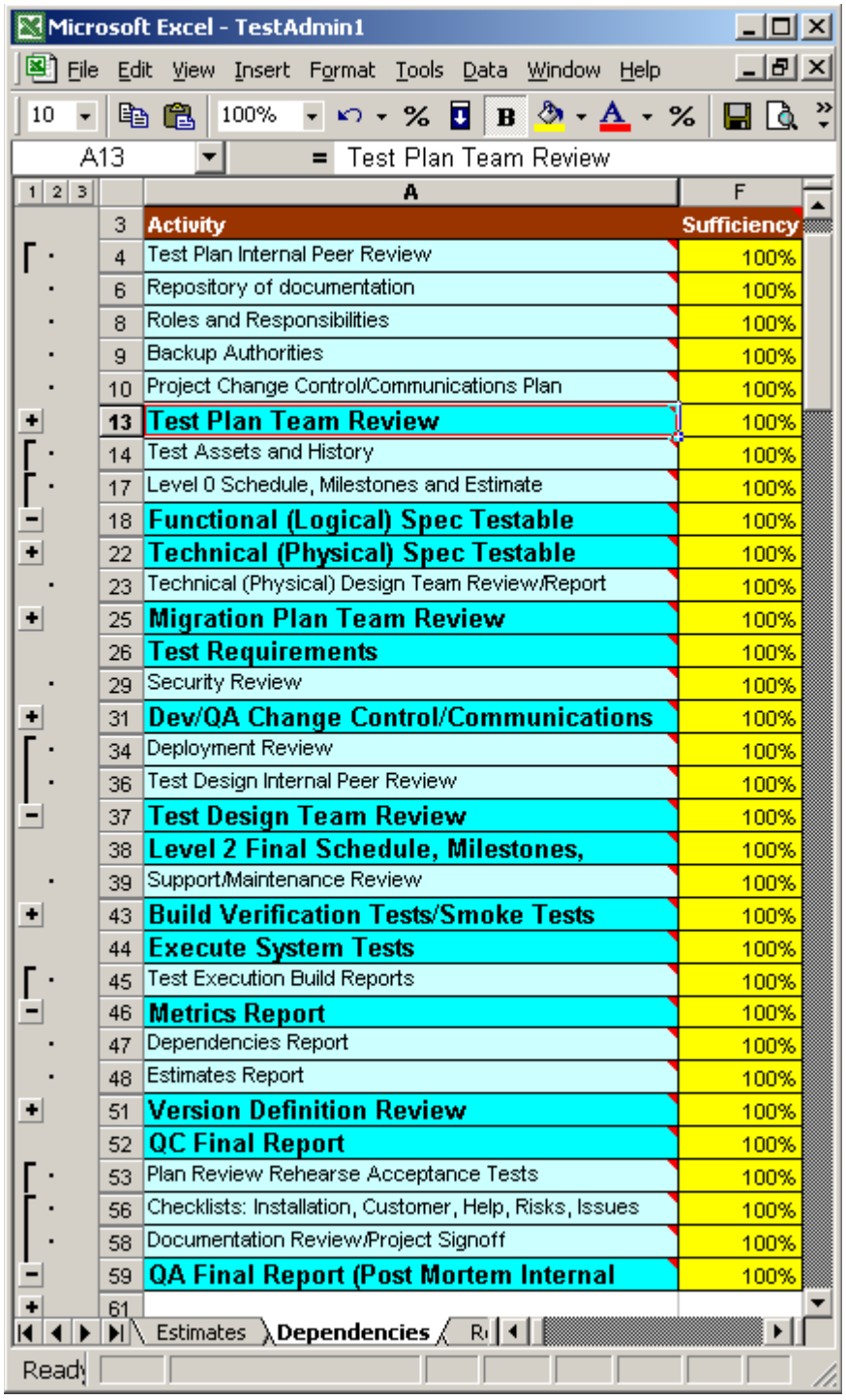
Project Sufficiency Rating:

QA rated process activities deliverables 30% less sufficient for evaluation and testing purposes in Project B than in Project A


Project A Project Sufficiency: 82%
120%
100%
80%
60%
40%
20%
0%
100%
85%
96%
79%
66%
Priority
Secondary
Other (Test Lead)
QA Tasks Addressed
Project Sufficiency


Project B Project Sufficiency: 57%
100.00%
80.00%
60.00%
40.00%
20.00%
0.00%
100%
63%
98%
52%
84%
Priority
Secondary
Other (Test Lead)
QA Tasks Addressed
Project Sufficiency

Baseline for this group:



	Activity	Sufficiency
4	Test Plan Internal Peer Review	100%
6	Repository of documentation	100%
8	Roles and Responsibilities	100%
9	Backup Authorities	100%
10	Project Change Control/Communications Plan	100%
13	**Test Plan Team Review**	100%
14	Test Assets and History	100%
17	Level 0 Schedule, Milestones and Estimate	100%
18	**Functional (Logical) Spec Testable**	100%
22	**Technical (Physical) Spec Testable**	100%
23	Technical (Physical) Design Team Review/Report	100%
25	**Migration Plan Team Review**	100%
26	**Test Requirements**	100%
29	Security Review	100%
31	**Dev/QA Change Control/Communications**	100%
34	Deployment Review	100%
36	Test Design Internal Peer Review	100%
37	**Test Design Team Review**	100%
38	**Level 2 Final Schedule, Milestones,**	100%
39	Support/Maintenance Review	100%
43	**Build Verification Tests/Smoke Tests**	100%
44	**Execute System Tests**	100%
45	Test Execution Build Reports	100%
46	**Metrics Report**	100%
47	Dependencies Report	100%
48	Estimates Report	100%
51	**Version Definition Review**	100%
52	**QC Final Report**	100%
53	Plan Review Rehearse Acceptance Tests	100%
56	Checklists: Installation, Customer, Help, Risks, Issues	100%
58	Documentation Review/Project Signoff	100%
59	**QA Final Report (Post Mortem Internal**	100%

Step 3: Identify the root cause of insufficiencies (ie. Communication problems, inappropriate skill set, insufficient hardware in the dev environment etc.)

Root Cause Analysis

What activities were not addressed by either projects

Code Review
Unit Test Plan Review
Client Application Recovery Plan
Execute and Report Acceptance Tests
Post Implementation Survey
Post Mortem Team Review

What activities were addressed in Project A that were not addressed in Project B:

Project Plan Review
Visible Project Schedule
Compliance: Business Rules
Standards: Dev Lifecycle
Requirements Analysis (Testable Statements) Internal Review
Requirements Team Review/Report
Req-Functional Spec Traceability Verification
Level 1 Schedule, Milestones, Estimate adjustment
Req-Functional-Physical Spec Traceability Verification
Architecture Review
Code Review
Help/Training Review
Risk Mitigation Review
Business Continuity Review
Deployment Review
Beta Communications Plan Review
Support/Maintenance Review
Unit Test Plan Review
Plan Review Rehearse Usability Tests
Execute and Report Usability Tests
Traceability Report
Client Application Recovery Plan (NSD Application Test Plan)
Business Customer Acceptance authorization

Of these, which could not be disputed as Evaluator error:

Project Plan Review
Visible Project Schedule
Standards: Dev Lifecycle
Requirements Analysis (Testable Statements) Internal Review
Requirements Team Review/Report
Req-Functional Spec Traceability Verification

Level 1 Schedule, Milestones, Estimate adjustment
Req-Functional-Physical Spec Traceability Verification
Plan Review Rehearse Usability Tests
Execute and Report Usability Tests
Traceability Report
Client Application Recovery Plan (NSD Application Test Plan)
Acceptance Evaluation Survey

Of these, which are most critical as Test Dependencies (Priority Activities)

Project Plan Review
Requirements Analysis (Testable Statements) Internal Review

Step 4: Work with manager to begin raising awareness of baseline with their team.

Conclusions

Metrics seem to support the conclusion that insufficiencies in these two Test critical areas: **Project Plan Review** and **Requirements Analysis,** appear to be chiefly responsible for most of the project problems for the same group in Project 2.

This conclusion was supported in Post Mortem discussions and also noted in the Final QC report for Project B.

5) Implement controls to raise insufficiencies to baseline levels.

Management of Build Migration from dev to Test, misunderstanding over the timing of System Integration activities, Changing User Requirements, and lack of Business documentation accounted for 67% of all recorded issues, and were identified as targets for improvement in the next project.

This analysis method could be adopted and used by management to:

- Determine management's Project process quality tolerance
- Determine management's Product quality tolerance
- Optimize project efficiency
- Control cost and schedule overruns

**This method uses standard desktop MS Office resources (MS Excel) and therefore requires NO investment in vendor Requirements or Project Management tools or resources.*

Implementation Requirements: Time frame 12 months

Appoint a QA resource as the Department Metrics Trainer and Analyst (65% resource time)
Endorse the use of the Beyond the Box Method and Templates in pilot involving all Project groups
Empower QA for access to required project information (Project hours other than QAs, authorization emails, etc.)
Train QAs in the use of the template and method. This would mean hands on, on the job instruction and monitor by the trainer
Baseline 1 project group over 6 months
Analyze and compare project findings.
Implement tolerance levels for New and Maintenance projects. This might mean a stated sufficiency level for all department projects, a stated sufficiency level for each group, a stated sufficiency level for Business Requirements, an acceptable percentage for cost overruns or calendar slip, or an acceptable ratio of process to project issues etc…
Measure Projects against department tolerance level or against each group's individual baseline
Determine fitness of method as a published Department standard.
Note: If also the tester, the QA evaluator should have outside QA peer or QA management oversight

Notes:

- IF the Evaluator is also a Tester on a project, oversight by the Trainer is required.

Disclaimer

- This presentation is EXPLICITLY NOT meant to single out groups or individuals at fault. It is meant to focus management's attention on areas in project and product attributes that appear insufficient for management's tolerance level.

- The Reader is reminded that there are always many hidden factors—both internal and external to any project--that could account for problems and issues that arise during the course of a project. These might include the team makeup, interpersonal styles of work, conduct and communication, changes to department priority, adequate human and materials resources, and participant task skill and experience level. This method should be exclusively used to focus management's attention to areas that might require deeper inquiry.